Developing a Quantum Key Distribution Simulator

May 2013

Sam Aybar, Richard Harrison, Inti Sairitupa, Jay Thomas, and Ron Frank
"Those who are not shocked when they first come across quantum theory cannot possibly have understood it."

- Niels Bohr
  (Nobel Prize Winner, Physics)
What is our Project?

Take a java-based Quantum Key Generator Simulation and Improve It

1. Improved GUI interface for inputting the data
   - Initial setup data
   - Filter/bit choice
   - Running bit/photon data

2. Improved browser-based reporting and formatting of errors and successful conclusion.
   - Distinguish runs with an Eve and without.
   - Distinguish throw-away bits from final bits

3. Documentation should comprise two documents.
   - An End-User Manual describing in detail how to set up and run the program and set up and run simulations (program use).
   - System Documentation with sufficient detail for maintenance and improvements.
Legacy Solution

```
java QuantumKeyDistributionSimulation "|/-" "EVE"
```

---

**EVE**

---

**ALICE**

---

**BOB**

---

**EVE**

---

```
Eve's RANDOM Filter Choices.
Vector Name = FilterSets:
[|+|] = Y
[|+] = N
[|+] = N
[|+] = Y
[|+] = Y

Eve's filter sets.
Vector Name = eveFilterSets:
[|+|] = X
[|+] = Y
[|+] = X
[|+] = X

Eve sends out Alice's (MODIFIED) photons to BOB. If she gets a gift (NO PHOTON) she passes on the known photons without a random choice.

Eve replaces missing photons with known photons and passes them on to Bob as if they were coming from Alice.

**BOB**

---

Bob's RANDOM Filter Choices.
Vector Name = filters:
[|+|] = Y
[|+] = N
[|+] = N
[|+] = Y
[|+] = X

---

**ALICE**

---

These are Bob's filter sets matching Alice's.

```
Definitions

- filter: device to orient photon in one of four "directions"
- filter set: each filter is part of a pair, matched with a filter 90 degrees different
Background on Encryption

• Founded on a conjecture
• Ultimately computational burden too large for an enemy.
• BUT: Faster Computers Coming
• BUT: Quantum Computers Will Break It!
What is Quantum Key Distribution?

- better method for exchanging encryption key
- uses 1-time pad - unbreakable
- solves the distribution problem
- detects if there is an eavesdropper
Measuring Quantum Bits (qubits)

Source: Dr. Ronald Frank
No Cloning Theorem

"Any attempt to measure (read) an unknown (mixed) state MUST modify (Project) that state."

Result: someone intercepting quantum transmissions cannot send it on unmodified.

Implication: with sufficient number of bits, you will ALWAYS know if there is an Eve
QKD Communication
The QKD Algorithm

Source: Dr. Ronald Frank
Our Solution

2 Versions

- Server based -- navigate to URL
- Installed -- run locally on machine
System Overview - Server Option

Web Front End

Mobile Front End

Internet

Broker

Back End

Soap

Core

Existing Code

Mysql
Backend Components - Server

- Backend Server
  - Core
    - Existing code refactored as well as core interfaces
  - Broker
    - Interface between the backend and web and mobile front ends
    - Also Provides Encryption / Decryption testing using the keys
  - Database
    - Storage of test results (both single and test suites) and real time simulations
System Overview - Local Option

Web Front End →

Java Archive (JAR)

Jetty

Externally Facing Servlets

Existing Code

SQLite Local Data
Stand Alone Components

- Stand Alone
  - Jetty
    - Lightweight Servlet Engine
    - Existing code refactored as well as core interfaces
  - Externally Facing Servlets
    - Interface between the backend and web and mobile front ends
    - Simulation of the encryption process for messages using a QKD distributed key
  - Local SQLite Datastore
    - Storage of test results (both single and test suites) and real time simulations
Front End Components

Web Front End

- Website is build in HTML 5.
- User can run QKD simulations by inputting data into a form.

Mobile Front End

- The website is liquid written in CSS and JavaScript
- Viewable on mobile device.
Simulator Use

Quantum Encryption
Using the properties of Quantum Physics for secure data transmission

Automatic Key Selection

On this page you can simulate Quantum Key Distribution using the number of bits you desire. Please enter a number in the box below to determine the length of the transmission between Alice and Bob. The simulator will randomly select the filters that Alice uses to polarize the photons and that Bob uses to read the photons. By checking the “Eve” box you can include an eavesdropper (whose filters will also be randomly generated) to see the effect on the simulation. Click “Run Simulation” to see the results.

Enter Filter Length: 

Eve: 

Run Simulation
Simulation Results

<table>
<thead>
<tr>
<th>Bit #</th>
<th>Alice's Filters</th>
<th>Alice's Filter Sets</th>
<th>Bob's Filters</th>
<th>Bob's Filter Sets</th>
<th>Eve's Filters</th>
<th>Eve's Filter Sets</th>
<th>Eve's Modified Filters</th>
<th>Eve's Modified Filter Sets</th>
<th>Bob and Matching Alice's Sets</th>
<th>Final Bits Received</th>
<th>Final Bit Matches</th>
<th>Final Bit#</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>\</td>
<td>X</td>
<td>1</td>
<td>\</td>
<td>X</td>
<td>\</td>
<td>X</td>
<td>Y</td>
<td>1</td>
<td>Y</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>\</td>
<td>X</td>
<td>1</td>
<td>-</td>
<td>+</td>
<td>\</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>N</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>+</td>
<td>0</td>
<td>\</td>
<td>+</td>
<td>/</td>
<td>X</td>
<td>\</td>
<td>X</td>
<td>Y</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>\</td>
<td>X</td>
<td>1</td>
<td>\</td>
<td>X</td>
<td>\</td>
<td>X</td>
<td>Y</td>
<td>1</td>
<td>Y</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>\</td>
<td>X</td>
<td>1</td>
<td>\</td>
<td>X</td>
<td>\</td>
<td>X</td>
<td>Y</td>
<td>1</td>
<td>Y</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>/</td>
<td>X</td>
<td>0</td>
<td>\</td>
<td>+</td>
<td>\</td>
<td>X</td>
<td>Y</td>
<td>1</td>
<td>N</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>\</td>
<td>X</td>
<td>1</td>
<td>\</td>
<td>X</td>
<td>\</td>
<td>X</td>
<td>Y</td>
<td>1</td>
<td>Y</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>\</td>
<td>X</td>
<td>1</td>
<td>\</td>
<td>+</td>
<td>+</td>
<td>\</td>
<td>X</td>
<td>\</td>
<td>Y</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>+</td>
<td>0</td>
<td>-</td>
<td>+</td>
<td>\</td>
<td>X</td>
<td>\</td>
<td>X</td>
<td>Y</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>\</td>
<td>X</td>
<td>1</td>
<td>\</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>Y</td>
<td>0</td>
<td>N</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Eve was Detected
Simulation Suite Results

**Suite Statistics:**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Simulations</td>
<td>1000</td>
</tr>
<tr>
<td>Filter Length</td>
<td>30</td>
</tr>
<tr>
<td>Average Filter Set Matches</td>
<td>14.999</td>
</tr>
<tr>
<td>Average Eve Impacted Bits</td>
<td>3.747</td>
</tr>
<tr>
<td>Number of Times Eve Undetected</td>
<td>9</td>
</tr>
</tbody>
</table>
Opportunities for Extension

1. Tweak QKD assumptions
2. Provide additional analytics
3. API integration to other systems
4. Native Mobile Clients
5. Build out QKD educational website
Demo